Sensor Placement for Classifier-Based Leak Localization in Water Distribution Networks using Hybrid Feature Selection

Adrià Soldevila[a,*], Joaquim Blesa[b], Sebastian Tornil-Sin[a,b], Rosa M. Fernandez-Canti[a], Vicenç Puig[a,b]

[a] *Research Center for Supervision, Safety and Automatic Control (CS2AC). Rambla Sant Nebridi, s/n, 08022 Terrassa (Spain).*
[b] *Institut de Robòtica i Informàtica Industrial (CSIC-UPC). Carrer Llorens Artigas, 4-6, 08028 Barcelona (Spain).*

Abstract

This paper presents a sensor placement approach for classifier-based leak localization in water distribution networks. The proposed method is based on a hybrid feature selection algorithm that combines the use of a filter based on relevancy and redundancy with a wrapper based on genetic algorithms. This algorithm is applied to data generated by hydraulic simulation of the considered water distribution network and it determines the optimal location of a prespecified number of pressure sensors to be used by a leak localization method based on pressure models and classifiers proposed in previous works by the authors. The method is applied to a small-size simplified network (Hanoi) to better analyze its computational performance and to a medium-size network (Limassol) to demonstrate its applicability to larger real-size networks.

Keywords: Sensor placement, leak localization, water distribution networks, feature selection, genetic algorithms, classifiers.

*Corresponding author.
Email address: adria.soldevila@upc.edu (Adrià Soldevila)

1. Introduction

Water Distribution Networks (WDNs) are critical infrastructures that need to be monitored to guarantee their satisfactory operation. One of the most common and critical issues to monitor and deal with are water leaks, which are account up to 30 % of the total amount of extracted water [1].

Water utilities have the common practice to divide the WDN into small and non-connected areas, called District Metered Areas (DMAs), to allow a better leak monitoring and pressure control, where the inlets are monitored with flow and pressure sensors and also a few pressure sensors are placed inside. Leak localization methods rely on the use of measurements provided by a set of installed sensors. Pressure sensors are normally preferred over flow sensors because they are cheaper and easier to install and maintain.

Several leak localization methods have been proposed in the literature, such as transient analysis, parameter estimation techniques, leak sensitivity analysis, mass-balance and linear programming algorithms [2], statistical interval estimation [3] and artificial intelligence based methods. Artificial intelligence techniques seem to be suitable tools to use since the problem to be solved presents several types of uncertainties. For instance, in [4] genetic algorithms are proposed to solve an optimization problem for simultaneously quantifying and locating water losses. In [5], a method based on the use of Support Vector Machines (SVM) is proposed that analyzes data obtained by a set of pressure control sensors of a pipeline network to locate and compute the size of a possible leak present in a WDN. More recently, the use of k-Nearest Neighbors (k-NN), Bayesian and neuro-fuzzy classifiers for leak localization purposes has been proposed in [6], [7] and [8].

Even for pressure sensors and due to budget constraints, the number of sensors that can be installed in practice is really limited. In this situation, the problem of sensor placement, i.e. the determination of the best locations inside the network to install the limited number of allowed sensors, is of utmost importance. Sensor placement in WDN was initially focus on water quality monitoring and it is still an active area of research [9, 10, 11] but in the last years some sensor placement methodologies for leak localization purposes have been proposed. Examples of these sensor placement methods are presented in [12], where an efficient branch and bound search is used, in [13, 14], where GAs are used, and in [15], where a prior clustering process is applied. In general, a given sensor placement method is designed for a particular leak localization method, since there is not a unique optimal set

of sensors for a given network (a set of sensors can be optimal for a given leak localization method but not for a different one).

In previous works [6], [7], the authors have proposed a framework for leak localization based on computing pressure residuals, i.e. differences between measurements provided by installed sensors and estimations computed from a normal-operation model of the network, and analyzing them by a classifier. In particular, the use of the k-NN classifier is presented in [6], whereas the use of a Bayesian classifier is proposed in [7]. In these works, it is assumed that there exist a small number of pressure sensors that are already installed in some internal nodes of the network. In this paper, it is assumed that the number of pressure sensors to install is given and the aim is to determine their optimal locations, i.e. the ones that maximize the leak localization performances obtained when the framework proposed in [6] and [7] is applied.

In this work, the problem of sensor placement is formulated as a Feature Selection (FS) problem. Feature (or variable/attribute) selection techniques [16] are used to identify a subset of relevant variables in a data set, regarding its use to build a model with a given purpose, for instance a classifier. Withing the framework proposed in [6] and [7], the main idea is to generate, using a hydraulic simulation of the considered WDN, a complete data set containing all the potential residuals associated to the network nodes and apply a FS algorithm that determines the ones that after training the classifier will provide the best leak localization results.

There are four main categories of FS techniques recognized in the literature [17, 18]: filter based methods, wrapper methods, embedded methods and, finally, hybrid methods, i.e. combination of filters with wrappers. The methods of the first type, filter based methods [19], directly work with the data, without interacting in any way with the model to be built. Hence, individual features or feature sets are evaluated according to some metrics that are assumed to be fast to compute. Some of the most common indicators are the relevance, i.e, the information contained in a given feature (according to the final application) [16, 20], and the redundancy, i.e. how much of the information in a given feature is repeated in others [21, 22]. Many existing filter methods combine these two indicators [23, 24]. The main advantage of this type of methods is their low computational cost, while the main drawback is that the selection does not take into account the posterior use of the data by the model. The second type of methods, wrapper methods [20], build and use the model to score selected feature subsets that are generated within the framework of an heuristic search. Some methods in this category are

based on the use of Genetic Algorithms (GAs) [25] and on Particle Swarm Optimization (PSO) methods [26], among others. Due to the search and to the fact that a new model has to be trained (build) for each subset, these methods are computationally demanding, but they usually provide the best results for the particular type of model used. Embedded methods are the third type of methods, and they combine the use of the model that ranks the features in a priority order to be selected. In this group, there are techniques such as Backward Feature Selection (BFS) [16], Random Forest (RF) [27] and, in general, Evolutionary Algorithms (EA) [28]. Finally, the most recent approaches are the hybrid methods, which typically combine a filter that reduces the initial number of features with a wrapper that provides an additional refinement [29, 30, 31]. The latter approach is considered in the present work due to the obtained good compromise between optimality and computation time.

According to the previous discussion, the contribution of this paper is the proposal of a sensor placement approach for classifier-based leak localization in water distribution networks that uses a particular hybrid feature selection algorithm, designed to reduce the computation time (for real WDNs with thousands of nodes the required computation time could be days or even weeks) while maintainig the (sub)optimality of the obtained results.

The rest of the paper is organized as follows. Section 2 presents the background, reviewing the architecture and methodology for leak localization based on pressure residuals and classifiers originally presented in [6] and [7]. Section 3 presents the formulation of the sensor placement problem as a feature selection problem. Section 4 details the proposed feature selection algorithm, which implements a hybrid method that uses a filter based on relevancy and redundancy/distance indicators and a wrapper based on a genetic algorithm. Section 5 presents the application of the proposed method to two networks of small and medium size: the simplified Hanoi WDN and a DMA of the Limassol WDN. Finally, Section 6 draws the main conclusions of the work.

Nomenclature

The names for the main variables and parameters used through the paper are summarized in Tables 1 and 2.

Table 1: Nomenclature for physical variables.

n_n	number of consumer nodes
$\tilde{d}_{WDN}$	measured global demand
d_i, $\hat{d}_i$ and $\bar{d}_i$	actual, estimated and generated demand in node i
$\mathbf{c}$, $\tilde{\mathbf{c}}$ and $\bar{\mathbf{c}}$	actual, measured and generated boundary conditions
$\tilde{\mathbf{p}}$ and $\hat{\mathbf{p}}$	measured and estimated inner pressures
l_i, $\hat{l}_i$ and $\bar{l}_i$	actual, estimated and generated leaks at node i
$\mathbf{v}$ and $\bar{\mathbf{v}}$	actual and generated noises

Table 2: Nomenclature for classifiers and feature selection.

n_c	number of classes of each feature
n_f and $n_f^{(R)}$	original and reduced number of features
n_s	number of features to be selected (inner pressure sensors to be installed)
n_b	number of fixed additional features (measured boundary conditions)
m_T and m_V	number of instances (examples) in each class in the training and validation data sets
$\mathbf{F}$ and $\mathbf{F}^{(R)}$	original and reduced features space
$\mathbf{T}$ and $\mathbf{T}^{(R)}$	original and reduced training data set
$\mathbf{V}$ and $\mathbf{V}^{(R)}$	original and reduced validation data set
$\mathbf{\Gamma}$	confusion matrix
$\mathbf{D}$	topological distance matrix
$\mathbf{\Phi}$ and $\mathbf{\Phi}^{(B)}$	original and binarized feature distance matrix
α	average value of the $\mathbf{\Phi}$ matrix except the diagonal values
σ	user defined threshold
γ	user defined threshold
$\mathbf{\Lambda}$	average training matrix
p_s	population size
e_c	elite count parameter
tol	fitness function tolerance
max_g	maximum number of generations

2. Background: Leak localization based on pressure residuals and classifier

2.1. Architecture and operation

In a previous work [32], the authors proposed an on-line leak localization method that relies on the scheme depicted in Fig. 1, based on computing pressure residuals and analyzing them by means of a classifier. Residuals are computed as the differences between the measurements provided by pressure sensors installed inside the DMA and estimations provided by a hydraulic model simulated under leak-free conditions. The WDN model is built using a hydraulic simulator such as Epanet and it is assumed to be able, after the corresponding calibration process using real data, to represent accurately the WDN behavior. However, in practice, nodal demands $(d_1, \cdots, d_{n_n})$ are not measured, except for some particular consumers where Automatic Metering Readers (AMRs) are available. So, it must be noted that the model is fed instead with estimated water demands $(\hat{d}_1, \cdots, \hat{d}_{n_n})$, typically obtained by the total measured DMA demand $\tilde{d}_{WDN}$ and distributed at nodal level according to historical consumption records in the nodes. Hence, the residuals are not only sensitive to leaks but also to differences between the real demands and their estimated values. Additionally, pressure measurements are subject to the effect of sensor noise $\mathbf{v}$ and this also affects the residuals. Taking all of these effects into account, the classifier must be able to locate the real leak present in the DMA, which can be located in any node (it is considered that the leaks can only be located in the nodes, and only one at a time) and with any (unknown) magnitude, while being robust to the demand uncertainty and the measurement noise. Finally, it must be noted that the operation of the network is constrained by some boundary conditions $\tilde{\mathbf{c}}$ (for instance the position of internal valves and reservoir pressures and flows) that are known (measured). These conditions are taken into account in the simulation and can also be used as inputs for the classifier.

2.2. Data generation

The application of the architecture described above (Fig. 1) relies on an off-line work whose main goal is to train and validate a classifier able to distinguish the potential leaks under the described uncertainty conditions. In this process, the data generation stage is critical. Since the data available from the real monitored WDN can be really limited, the way to obtain a complete training data set is by using a hydraulic simulator. Hence, training

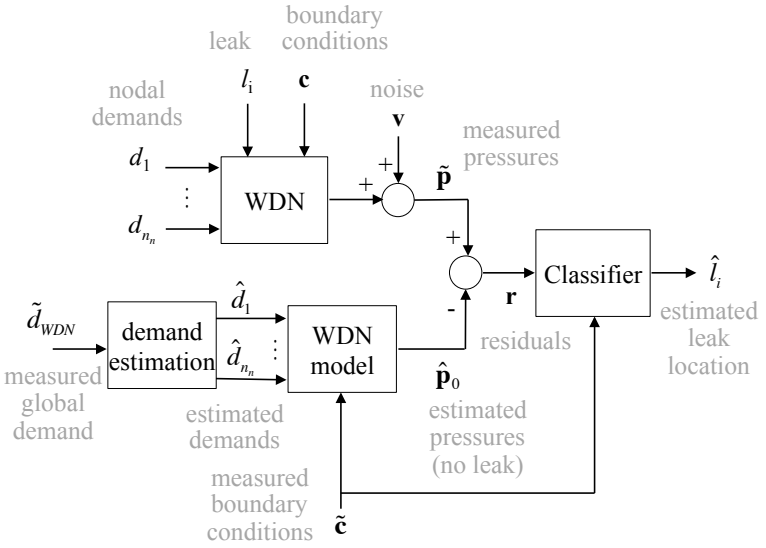

Figure 1: Leak localization scheme.

(and also validation and testing) data are generated by applying the scheme depicted in Fig. 2, similar to the one presented in Fig. 1 but with the main difference of substituting the real WDN by a model that allows to simulate the WDN not only in normal operation but also in presence of leaks.

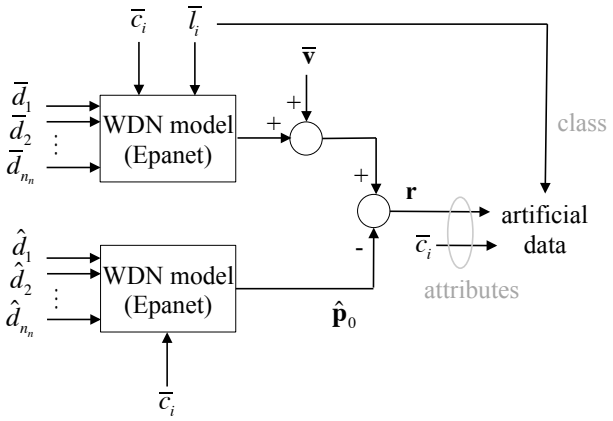

Figure 2: Data generation scheme.

The presented scheme is used to:

- Generate data for all possible leak locations, i.e. for all the different nodes in the WDN ($\bar{l}_i, \quad i = 1, 2, ..., n_n$).
- Generate data for different leak magnitudes inside a given range ($\bar{l}_i \in [l_i^-, l_i^+]$), for each possible leak location.
- Generate sequences of demands ($\bar{d}_1, ..., \bar{d}_{n_n}$) and boundary conditions $\bar{\mathbf{c}}$ that correspond to realistic typical daily demand evolution in each node.
- Simulate differences between the real demands and the estimations computed by the demand estimation module $\left((\bar{d}_1, ..., \bar{d}_{n_n}) \neq (\hat{d}_1, ..., \hat{d}_{n_n})\right)$.
- Take into account the measurement noise in pressure sensors, by generating synthetic Gaussian noise ($\bar{\mathbf{v}}$).

Taken into account the goal of this paper, it must be highlighted that the models compute the internal pressures in all the network nodes and that the presented data generation scheme allows generating a complete data set that can be analyzed to determine which pressure measurements are more useful for leak localization purposes.

2.3. Classifier evaluation

To evaluate the trained classifier, the confusion matrix $\mathbf{\Gamma}$ can be computed, which summarizes the results obtained when the classifier is applied to a validation (or testing) data set. Applied to the leak localization problem and using the associated terminology, the confusion matrix is a square matrix with as many rows and columns as classes in the classifier, which is equal to the number of nodes in the network (potential leak locations), where each coefficient $\Gamma_{i,j}$ indicates how many times a leak in node i is recognized as a leak in node j. Table 3 illustrates the concept of the confusion matrix applied to leak localization.

In case of a perfect classification, the confusion matrix should be diagonal, with $\Gamma_{i,i} = m_V$, for all $i = 1, \cdots, n_c$ being m_V the size of the validation data set (number of examples in each class) and n_c the number of classes. In our case, according to the previous section, the number of classes n_c is the number of places of the WDN where a leak is considered, i.e. the number of

Table 3: Confusion matrix $\mathbf{\Gamma}$

	$\hat{l}_1$	$\cdots$	$\hat{l}_i$	$\cdots$	$\hat{l}_{n_c}$
l_1	$\Gamma_{1,1}$	$\cdots$	$\Gamma_{1,i}$	$\cdots$	Γ_{1,n_c}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l_i	$\Gamma_{i,1}$	$\cdots$	$\Gamma_{i,i}$	$\cdots$	Γ_{i,n_c}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
l_{n_c}	$\Gamma_{n_c,1}$	$\cdots$	$\Gamma_{n_c,i}$	$\cdots$	Γ_{n_c,n_c}

network nodes n_n. But this number could be changed for example removing the data from nodes that are very close to others and reducing the number of potential leaks (classes).

In practice, non-zero coefficients will appear outside the main diagonal of matrix $\mathbf{\Gamma}$. For a leak in node i, the coefficient $\Gamma_{i,i}$ indicates the number of times that the leak l_i is correctly identified as $\hat{l}_i$, while $\sum_{j=1}^{n_c} \Gamma_{i,j} - \Gamma_{i,i}$ indicates the number of times that it is wrongly classified. The overall accuracy Ac of the classifier is defined as

$$\mathrm{Ac} = \frac{\sum_{i=1}^{n_c} \Gamma_{i,i}}{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Gamma_{i,j}} \tag{1}$$

Since the accuracy value (1) only provides a reference of the classification goodness and not how good it is the leak localization, the Average Topological Distance (ATD) is the indicator used to assess the overall performance. The ATD is the average value of the the minimum distance in nodes between the node with the leak and the node candidate proposed by the leak localization method. The ATD is computed as follows

$$\mathrm{ATD} = \frac{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Gamma_{i,j} \mathrm{D}_{i,j}}{\sum_{i=1}^{n_c} \sum_{j=1}^{n_c} \Gamma_{i,j}} \tag{2}$$

where $\mathbf{D}$ is a symmetric square matrix with size n_c such that each element $\mathrm{D}_{i,j}$ contains the minimum topological distance in nodes between the nodes referred by i and j.

3. Sensor placement problem as a feature selection problem

The objective of this work is to develop an approach to place a given number of inner pressure sensors, n_s, in a DMA of a WDN in order to obtain

a sensor configuration with a maximized leak isolability performance when using the leak localization method presented in the previous section. This problem can be recast into a feature selection (also known as variable or attribute selection) problem. The solution of this problem aims at selecting a subset of relevant and non-redundant features (variables) for use in the classifier construction. A feature selection algorithm combines a search technique for proposing new feature subsets, along with an evaluation measure which scores the different feature subsets. The simplest algorithm is to test each possible subset of features finding the one which minimizes the error rate. However, this is an exhaustive search of the space that is computationally intractable except for small feature sets.

To select a configuration with n_s sensors (features), the following binary vector is defined

$$\mathbf{q} = \left(q_1, ..., q_{n_f}\right) \tag{3}$$

where n_f is the total number of features, $q_j = 1$ if the pressure in the node j is measured, and $q_j = 0$ otherwise (i.e. the vector $\mathbf{q}$ indicates which sensors are installed of all the possible ones). In this paper we will consider that it is possible to place a sensor in all the nodes of the network, i.e. $n_f = n_n$. But this number could be reduced to a subset of these possible places determined by the WDN management company. Moreover the n_s selected features (pressure residuals of inner nodes), as was described in previous section and as can be seen in Fig. 1, the classifier is fed with the measured boundary conditions of the network $\tilde{\mathbf{c}}$ that will provide n_b fixed additional features.

In order to evaluate the quality of a feature selection regarding the leak localization performance, the average topological distance index (2) will be used. This performance index depends on the configuration of features considered and it is parameterized in terms of the $\mathbf{q}$ vector (3) to determine the best selection

$$\mathrm{ATD}(\mathbf{q}) = \frac{\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}\Gamma_{i,j}(\mathbf{q})\mathrm{D}_{i,j}}{\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}\Gamma_{i,j}(\mathbf{q})} \tag{4}$$

Based on the vector $\mathbf{q}$ and the performance index $\mathrm{ATD}(\mathbf{q})$, the feature selection problem can be translated into an optimization problem formulated as follows

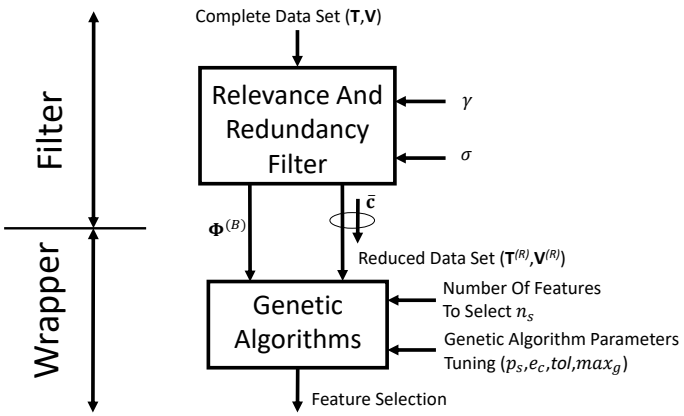

Figure 3: Hybrid feature selection scheme.

$$\begin{aligned} &\min_{\mathbf{q}} \mathrm{ATD}(\mathbf{q}) \\ &\text{s.t.} \\ &\sum_{j=1}^{n_f} \mathrm{q}_j = n_s \end{aligned} \tag{5}$$

where $\mathbf{q}$ is defined according to (3) and $n_s \in \{1, ..., n_f\}$ is the given number of sensors (features) to be selected.

4. Proposed solution: a hybrid feature selection algorithm

4.1. Overview

The proposed method is a hybrid approach with two different stages that are performed in a sequential way. First, an initial reduction of the dimension is performed by using a filter based on evidences of the relevancy and redundancy of the variables that additionally assesses information about suitable/unsuitable pairs of combinations of features. Second, the subset of features that remains after the filter and the additional information is taken into account by the proposed wrapper method, which is a genetic algorithm, to tackle the combinatory problem and obtain a suboptimal feature selection. The whole procedure is depicted in Fig. 3.

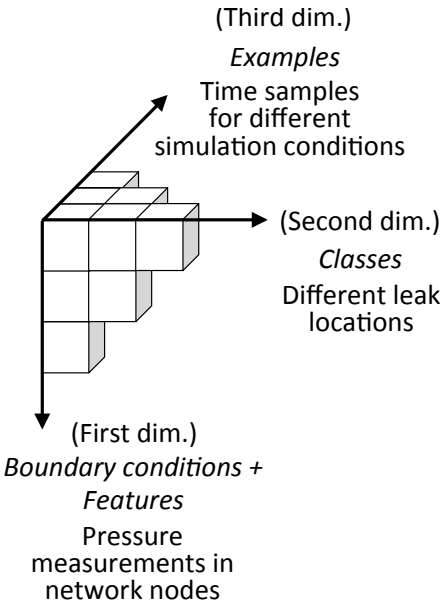

Figure 4: Data format.

4.2. Data format

The algorithms presented in the following subsections assume that the data generated after the simulation of the network under different conditions (see Section 2.2) are organized in a particular way. Hence, both the training $\mathbf{T}$ and validation $\mathbf{V}$ data matrices are three-dimensional matrices, as shown in Fig. 4. The first dimension of those matrices is associated to the features, i.e. to the n_b boundary measurements and the n_f pressure residuals computed at the different nodes of the network where pressure sensors can be installed. The second dimension is associated to the classes (with size n_c), i.e. to the different possible leak locations that are considered. Finally, the third dimension is associated to the examples: for a fixed residual and a fixed leak location, the third dimension collects residual values that are obtained under different leak sizes, node demand estimations, noise realizations and at different time instants. The length of this last dimension will be denoted as m_T and m_V for training and validation matrices.

4.3. Filtering

With the aim of reducing the computational load of the wrapper, an initial reduction of the n_f dimension of the original feature space $\mathbf{F}$ of inner pressure residuals is applied. This dimensionality reduction is based on

a relevance/redundancy-based filter that removes the most irrelevant and similar features. This reduction increases the performance of the proposed wrapper strategy which is based on the use of genetic algorithms. The proposed filter is based on the Fast Correlation-Based Filter (FCBF) presented in [23], where first the features are ranked based on their relevance and then a sequential procedure to remove the redundant (and less relevant) features is performed.

4.3.1. Relevance metric

Relevance is associated to the information that a given feature possesses according to the final problem to be solved. For classification problems, relevance is associated with the variability of the feature across the classes. Hence, in order to compute the relevance of each feature, the average training matrix $\mathbf{\Lambda}$ with size $n_f \times n_c$ and associated to inner pressure residuals is defined with features as rows and classes as columns,

$$\Lambda_{i,j} = \frac{1}{m_T} \sum_{a=1}^{m_T} \mathrm{T}_{(n_b+i),j,a} \quad i = 1, ..., n_f \quad j = 1, ..., n_c \tag{6}$$

where $\mathrm{T}_{(n_b+i),j,a}$ are the elements of the training matrix $\mathbf{T}$ associated to inner pressure residuals, obtained as described in Section 2.2, with size $(n_b + n_f) \times n_c \times m_T$ where all the instances are stored with features as rows, classes as columns and different instances (examples) in the third dimension.

It is considered that an indirect measure of the relevance of each feature, R_z for $z = 1, ..., n_f$, can be computed as follows

$$R_z = \frac{2}{n_c^2 - n_c} \sum_{i=1}^{n_c-1} \sum_{j=i+1}^{n_c} (\Lambda_{z,i} - \Lambda_{z,j})^2 \tag{7}$$

4.3.2. Redundancy metric

As an indirect measure of the redundancy of $\mathbf{F}$, the similitude or proximity degree between each pair of features is used. In [33], it was proposed to use row vectors of the leak sensitivity matrix to measure the similitude between the behaviour of two inner pressure sensors in the presence of the different leak scenarios. In this work, we propose to use the 2-norm between the average values of each possible pair of features. Then, considering the row vectors of the matrix $\mathbf{\Lambda}$

$$\mathbf{\Lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_{n_f} \end{pmatrix} \tag{8}$$

where

$$\boldsymbol{\lambda}_i = (\Lambda_{i,1}, ..., \Lambda_{i,n_c}) \quad i = 1, ..., n_f, \tag{9}$$

a feature distance matrix $\mathbf{\Phi}$ can be defined such that its components store the measure of the redundancy between each pair of features i and j and are computed as

$$\Phi_{i,j} = \|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j\|_2 \qquad \forall i = 1, ..., n_f \text{ and } j = 1, ..., n_f \tag{10}$$

Matrix $\mathbf{\Phi}$ is a symmetric square matrix of dimension n_f where all the diagonal elements are 0, thus indicating that each feature presents zero distance to itself, in other words, each feature is totally redundant to itself.

4.3.3. Filtering process

The filtering process starts by computing the relevance for all the features, R_z, $z = 1, ..., n_f$, according to (7). The computed values are introduced in the relevance vector $\mathbf{R}$ and they are sorted in descending order in $\mathbf{R}_R$. On the other hand, the feature distance matrix $\mathbf{\Phi}$, that stores the distance between pairs of features is computed, according to (10).

The core of the algorithm is an iterative process that starts by considering the feature corresponding to the first value of $\mathbf{R}_R$, i.e. the most relevant feature. This feature is first compared in terms of similitude with the next feature in the relevance ranking. Taking into account the associated coefficient in $\mathbf{\Phi}$ and a user defined threshold γ, if the distance between the two considered features is lower than the threshold then the second (and less relevant) feature is removed from the feature space $\mathbf{F}$. The comparison and elimination process is repeated until the most relevant feature has been compared with all the other features in the list. And the whole process already applied to the first feature is repeated for the rest of features in the list. At the end, the feature space with the remaining features $\mathbf{F}^{(R)}$ is obtained, being its dimension $n_f^{(R)}$.

Finally, the filtering process ends with the computation of a matrix that will be used in the wrapper stage. According to a new user defined threshold

σ, a binary square matrix $\mathbf{\Phi}^{(B)}$ of dimension $n_f^{(R)}$ is defined. This matrix collects the information of which pairs of features of the remaining $n_f^{(R)}$ features after the filtering stage are suitable to be combined or not in the same potential feature selection group in the wrapping stage according their dissimilarity. The components of this matrix are computed as

$$\Phi_{i,j}^{(B)} = \begin{cases} 0, & \text{if } |\Phi_{i',j'}| < \sigma \\ 1, & \text{if } |\Phi_{i',j'}| \geq \sigma \end{cases} \tag{11}$$

where the indices i and j are related with the indices i' and j' with a mapping function that maps the features of $\mathbf{F}^{(R)}$ in the features of the original feature space $\mathbf{F}$.

Notice that σ has to be bigger than γ to have an impact on the wrapper because pairs of features with feature distance smaller than γ are removed in the filtering stage. Both values for γ and σ can be expressed in a relative way with respect to the average of the coefficients of $\mathbf{\Phi}$ outside the main diagonal, denoted as α and computed as

$$\alpha = \frac{2}{n_c^2 - n_c} \sum_{i=1}^{n_c-1} \sum_{j=i+1}^{n_c} \Phi_{i,j} \tag{12}$$

The whole filter process is summarized in Algorithm 1.

4.4. Wrapper search

The wrapper used in the second stage of the hybrid feature selection proposed in this paper is a genetic algorithm (GA).

GA provide an optimization engine based on the genetic evolution, where starting from a seed population (first generation) with a fixed population size p_s, each member of the population is evaluated according to a fitness (or objective) function and ranked. The best ones (their number is determined by the elite count parameter e_c) survive to the next generation, and the remaining members of the new generation (until the p_s number) are members derived from the ones that have survived. The process is repeated until one of the stopping criteria is accomplished, for instance, the maximum number of generations max_g is reached, or no best member has been found from one generation to the next (i.e., the difference is less than a tolerance tol).

The members of the first generation of population are randomly created by the GA with the specified population size p_s and then evaluated to select the best ones. The next generation of population is obtained from the elite

Algorithm 1 Relevance and redundancy/distance filter.

Require: A features space $\mathbf{F}$ and their size n_f, a training matrix $\mathbf{T}$, the number of classes n_c of each feature, the number of instances in each class m_T in the training matrix and the user defined thresholds γ and σ.

Ensure: Remove the redundant (and less relevant) features below γ.

for $i = 1, \cdots, n_c$ **do**
  for $j = 1, \cdots, n_c$ **do**
    $\Lambda_{i,j} = \frac{\mathbf{1}}{\mathbf{m_T}} \sum_{a=1}^{m_T} \mathrm{T}_{(n_b+i),j,a}$
  end for
end for
for $z = 1, \cdots, n_f$ **do**
  $R_z = \frac{2}{n_c^2 - n_c} \sum_{i=1}^{n_c-1} \sum_{j=i+1}^{n_c} (\Lambda_{z,i} - \Lambda_{z,j})^2$
end for
Rank in $\mathbf{R}_R$ the features according to their position in $\mathbf{R}$.
for $i = 1, \cdots, n_f$ **do**
  for $j = 1, \cdots, n_f$ **do**
    $\Phi_{i,j} = \|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_j\|_2$
  end for
end for
for $i = 1, \cdots, n_f$ **do**
  if $R_{R,i} \geq 0$ **then**
    for $j = i + 1, \cdots, n_f$ **do**
      if $\Phi_{i,j} < \gamma$ **and** $R_{R,j} \geq 0$ **then**
        $R_{R,j} = -1$
      end if
    end for
  end if
end for
Removed all the features from the space $\mathbf{F}$ with negative argument in $\mathbf{R}_R$ to create the new reduced space $\mathbf{F}^{(R)}$ with $n_f^{(R)}$ number of features. Also create, using σ and (11), the binarized feature distance matrix $\mathbf{\Phi}^{(B)}$.

members of the previous one, which directly pass from one generation to the next, and a number of filling members created by a pool onto the former best members. These filling members are identified by means of a bit string and are chosen by a tournament selection (which has a default value of 2 in Matlab) with the application of Laplace crossover (fusion of two members by swapping parts of their bit strings, Matlab default value of 0.8) and power mutation (when a bit in the string that define the members is changed, Matlab default value of 0.1) with a truncation process to ensure integer members. For further details see [34].

Two modifications have been included into the basic GA scheme to increase its speed. On the one hand, the information of the objective function is stored in the case that the members that appeared earlier appear again. In such a case the stored value is retrieved instead of calculating the fitness function again (as proposed in [35]). On the other hand, the matrix $\mathbf{\Phi}^{(B)}$ is used to avoid fitness functions calculations. Hence, when a combination contains a not allowed pair of features, the worst value is directly assigned without the computation of the fitness function.

A new training matrix $\mathbf{T}^{(R)}$ is built by removing the features discarded by the filter. This matrix has dimension $(n_b + n_f^{(R)}) \times n_c \times m_T$. In a similar way, a new validation matrix $\mathbf{V}^{(R)}$ of $(n_b + n_f^{(R)}) \times n_c \times m_V$ dimension is created, where m_V is the number of instances of each class in the validation data set.

The pseudo-code of the algorithm is shown in Algorithm 2. First, the GA is initialized by adjusting the tuning parameters (line 1) which include the population size p_s, the bit string type population, the elite count e_c to maintain members between iterations, the tolerance *tol* and the maximum number of generations max_g to stop the optimization. Then, the constraints of the optimization variables are defined (line 2), which include the number of features to select n_s.

After that, the GA optimization process runs as an iterative process (lines 4 to 24), where the first step is to create the generation of members ($\mathbf{I}$ matrix) to be evaluated (line 6) and then evaluate them all (lines 7 to 22). Firstly, the algorithm checks (function GetUsed()) if the member ($\mathbf{q}$ vector) is new or repeated (line 9). If the combination is repeated the stored fitness value is retrieved (function GetATD()) (line 20) instead of computing the fitness function again, and the member evaluation finalizes, otherwise the process continues. The next step in the process is to check if the member is

an allowed combination of features according to the $\mathbf{\Phi}^{(B)}$ matrix (function CheckCombinations()) (line 10). If the combination is not allowed, then the fitness value is set to the worst result (line 17) and the member evaluation finalizes; otherwise, if it is allowed, then the process continues.

To perform the evaluation itself, first the classifier is created by using the training matrix $\mathbf{T}^{(R)}$ (line 11) where the $\mathbf{q}$ vector is used to select the adequate columns, and the confusion matrix $\mathbf{\Gamma}$ is obtained by using the classifier and the validation matrix $\mathbf{V}^{(R)}$ (line 12). Then, the fitness indicator is computed from the confusion matrix (line 13), the member is set to used (function SetUsed()) (line 14) and the fitness value is stored (function SetATD()) (line 15).

Finally, the best member of the generation is evaluated against the past ones, and it is checked if any of the required criteria to stop the optimization is reached (line 23).

5. Case studies

In this section, the proposed feature selection algorithm is applied to the sensor placement problem in two different water distribution networks. Firstly, it is applied to a simplified version of the Hanoi (Vietnam) WDN that allows to analyze the performance of the method by comparing the obtained results and associated computing time to the reference values provided by an exhaustive search. Second, the method is applied to a DMA of the Limassol (Cyprus) WDN, a medium-size network that is used to illustrate the applicability of the method to medium real-size networks.

5.0.1. Bayesian Classification

The sensor placement method proposed in this paper can be used for any classifier in the leak localization scheme of Fig. 1. Here, for illustration purposes, we use the Bayesian classifier proposed in [7]. This Bayesian classifier assumes that the distribution of each fault (leak location) is Gaussian distributed, and dependence between residuals can exist (i.e. the covariance matrix is not necessarily diagonal). To calculate the probabilities, the Bayesian classifier uses Probability Density Functions (PDFs) calibrated by using the training data set $\mathbf{T}^{(R)}$, where each leak location (l_i) has its own PDF.

So, the leak probabilities at each node are obtained by combining the observed residual $\mathbf{r}$ with the training data calibrated PDFs by means of the

Algorithm 2 Sensor placement based on Genetic Algorithms.

Require: A training matrix $\mathbf{T}^{(R)}$ and a validation matrix $\mathbf{V}^{(R)}$. A feature space $\mathbf{F}^{(R)}$ and their size $n_f^{(R)}$, the number of classes n_c, the number of features to select n_s, the population size p_s, the elite count parameter e_c, the fitness function tolerance tol, the maximum number of generations allowed max_g, the distance matrix $\mathbf{D}$ and the reduced binarized matrix $\mathbf{\Phi}^{(B)}$.

Ensure: A near-optimal sensors configuration $\mathbf{q}$ with error index $\text{ATD}_{\mathbf{min}}$.

$init \leftarrow \text{InitVarGA}(p_s, e_c, tol, max_g)$
$constraint \leftarrow \text{SetConstraints}(n_f^{(R)})$
Inputs: $init$, $constraint$, $\mathbf{T}^{(R)}$, $\mathbf{V}^{(R)}$, p_s, n_c, $\mathbf{\Phi}^{(B)}$.
while An optimization criterion is not reached **do**
  GA based search:
  Generate $\mathbf{I}$ matrix of size $(p_s \times n_f^{(R)})$ where each row is a member of a generation from the space $\mathbf{F}^{(R)}$.
  for $k = 1, \cdots, p_s$ **do**
    $q_k \leftarrow I_k$
    if $\text{GetUsed}(q_k) = 0$ **then**
      if $\text{CheckCombinations}(\mathbf{\Phi}^{(B)}, q_k) = 1$ **then**
        $\text{Classifier}(q_k) \leftarrow \text{Train}(\mathbf{T}^{(R)}(q_k))$
        $\mathbf{\Gamma}(q_k) \leftarrow \text{Validate}(\text{Classifier}(q_k), \mathbf{V}^{(R)}(q_k))$
        $\text{ATD}(q_k) \leftarrow \frac{\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}\Gamma_{i,j}(q_k)\text{D}_{i,j}}{\sum_{i=1}^{n_c}\sum_{j=1}^{n_c}\Gamma_{i,j}(q_k)}$
        $\text{SetUsed}(\mathbf{q}(k))$
        $\text{SetATD}(\text{ATD}(q_k, q_k)$
      else
        $\text{ATD}(q_k) = 0$
      end if
    else
      $\text{ATD}(q_k) = \text{GetATD}(q_k)$
    end if
  end for
  Find $\{\mathbf{q}, \text{ATD}_{\mathbf{min}}\}$ such that
    $\text{ATD}_{\mathbf{min}} = \min_{\mathbf{q}}(\text{ATD}(q_1, ..., \text{ATD}(q_{p_s}))$.
end while

application of the Bayes rule as

$$P(l_i \mid \mathbf{r}) = \frac{P(\mathbf{r} \mid l_i)P(l_i)}{P(\mathbf{r})},\ i = 1, ..., n_n \tag{13}$$

where $P(l_i \mid \mathbf{r})$ is the posterior probability that the observed residual vector $\mathbf{r} = (r_1, \cdots, r_{n_f}, \tilde{c}_1, \cdots, \tilde{c}_{n_b})^T$ is caused by the leak l_i, $P(\mathbf{r} \mid l_i)$ is the likelihood of the observed residual $\mathbf{r}$ assuming that the actual leak is l_i, $P(l_i)$ is the prior probability for the leak l_i, and $P(\mathbf{r})$ is a normalizing factor given by the Total Probability Law, defined as

$$P(\mathbf{r}) = \sum_{i=1}^{n_n} P(\mathbf{r} \mid l_i)P(l_i) \tag{14}$$

The prior probabilities are considered equally probable, that is,

$$P(l_i) = \frac{1}{n_n},\ i = 1, ..., n_n \tag{15}$$

The resulting posterior probabilities provide the probability for each leak l_i given the observed residual $\mathbf{r}$. Then, the leak with highest posterior probability is the most probable leak and is considered the node candidate (i.e. output of the classifier).

5.1. Hanoi WDN case study

The first case study is based on the network depicted in Fig. 5, a simplification of Hanoi city (Vietnam) WDN network with 31 nodes, 36 pipes and one reservoir. One flow sensor is placed in the inlet, and it is considered that the suitable number of pressure sensors to place inside the network is two, since that number offers an acceptable degree of leak isolation [13].

The considered sampling time for pressure measurements is 10 minutes, since this is a common value in real SCADA (Supervisory, Control And Data Acquisition) systems. Ten different data sets are generated with the following uncertainties created with uniformly distributed bounded random numbers with the aim of generate realistic scenarios (for more details about how these uncertainties are generated see [36]):

- Leak size in the range between 25 and 75 [l/s].
- Nodal demands with ± 10 % of uncertainty of the nominal value.

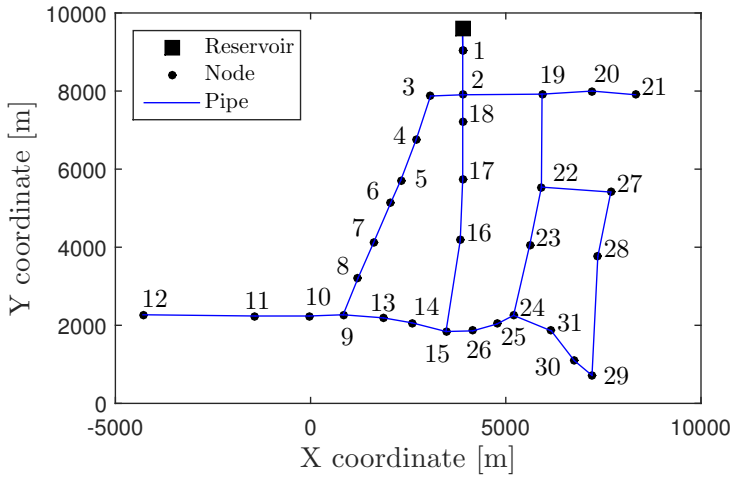

Figure 5: Hanoi DMA topological network.

- Noise in the measurements with a ±5 % of uncertainty of the average value of all pressure residuals.

Also the daily pattern consumption used is extracted from a real network (the average daily pattern consumption from several days is used) and scaled to the size of this network. From this unique daily pattern consumption ±12.5 % uncertainty of the nominal value is added using bounded uniformly distributed random numbers to obtain different consumption daily patterns like the ones depicted in Fig. 6.

A Bayesian classifier is fed with the training matrix $\mathbf{T}^{(R)}$ to calibrate the PDFs, which are considered to have Gaussian distribution (as justified in [37]). Ten different data sets (training and validation) are created to evaluate the different methods since the lower the data set size, the faster the computation is. But, since the GAs need to be executed several times to avoid local minima, changing the data set allows to avoid strange (due to the outliers) data sets.

Each data set is split into a training and validation data subsets. The training data subset consists in four days of data (96 samples for each class, i.e. leak location) and the validation data subset consists in two days of data (48 samples for each class). The training data subset is used in the filter stage

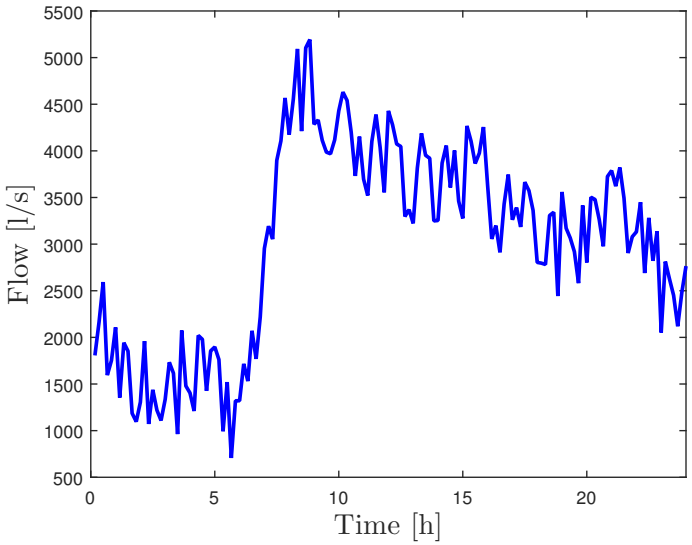

Figure 6: Example of a daily flow consumption in Hanoi DMA network.

to generate the subset of features $\mathbf{F}^{(R)}$ and in the wrapper stage to train the classifier for the current feature selection candidate. On the other hand, the validation data subset is used to assess the performance (ATD value) of the current feature selection candidate produced by the wrapper. To compare the different results obtained for the different data sets, a unique testing data set is used. This testing data set consists in 20 days of data (480 samples for each class) and is used to calculate the ATD value showed in Tables 4, 5 and 6.

For the small Hanoi DMA network, the Exhaustive Search (ES) method can be applied. The ES method guarantees the optimal solution by performing all the possible combinations among all the features, which is $\frac{n_f!}{n_s!\left(n_f-n_s\right)!}$, but usually this method is not suitable in terms of computational time. Given the small size of the Hanoi DMA network, and the constraint of only two sensors to be selected (n_s=2) among 31 possible places where pressure sensors can be installed (n_f=31), the ES method can be applied because the $\left(\frac{31!}{2!\,(31-2)!}\right) = 465$ possible combinations is a reasonable number to be evaluated exhaustively. The results obtained by the ES method are used to compare the efficiency of the final solution and computation time for the proposed approach. On the other hand, a standalone wrapper method that

Table 4: Results of the Exhaustive Search method in the Hanoi DMA network.

Data set	Sensors	Time ES	ATD
1	14, 28	573	1.21
2	11, 28	570	1.24
3	10, 27	579	1.13
4	**13, 27**	**569**	**1.09**
5	13, 27	570	1.11
6	13, 27	573	1.13
7	13, 29	571	1.13
8	13, 27	573	1.15
9	13, 28	568	1.14
10	13, 28	563	1.13
Average	-	571	1.15

only considers the proposed wrapper stage (Algorithm 2 without previous filtering nor the use of the matrix $\mathbf{\Phi}^{(B)}$) will be applied to illustrate the advantages in the sensor placement performance of the proposed hybrid method (filter+wrapper).

The results for the ES method in the Hanoi DMA network are summarized in Table 4, where the term "Sensors" refers to the selected node locations where the inner pressure sensors will be placed, "ATD" is the average topological distance in [nodes] computed by (2) using the testing data set, and "Time ES" is the time required to obtain the solution by the ES method in [s]. Regarding the computing time, it must be remarked that all the computations summarized in this paper have been performed by using a PC with an INTEL(R) CORE(TM) i7-4720HQ CPU @ 2.60 [GHz], 8 [GB] of RAM memory and a Windows 10 Home 64 bits OS and using the MATLAB 2015a software and the global optimization toolbox.

It can be noticed from Table 4 that different data sets lead to different results, i.e. to different pairs of selected sensors. This is due to the fact that there are sets of close nodes for which the behavior in terms of pressure is quite similar and due to the randomness of the data generation. Then, the results for one data set can include a given feature/node while the results for a different data set can include a different but close node. In Table 4, the nodes that appear as first selected node for all the 10 data sets are nodes 10, 11, 13 and 14, which can be identified as neighbors in Fig. 5. And the

same happens with nodes 27, 28 and 29 that appear as second selected node. The finally selected sensors are the pair $\{13, 27\}$, highlighted in bold in the table and obtained for the data set 4, because for this data set and this pair of sensors the obtained ATD is the minimal one. It must be noticed that the different combinations have a very close performance and all of them are good configurations to place the sensors, far better than the discarded configurations.

In the next experiment, a simplified version of the method that just implements the proposed GA wrapper stage has been tested. The following values have been used for the tunable GA parameters:

- Tolerance of $tol = 10^{-6}$ to determine that the optimal value is reached.
- Population size $p_s = 5$ (as it is shown in [38] that provides good results).
- Elite count of $e_c = 0.05p_s$, but at least one survives (which is the case given the selected p_s).
- The maximum number of generations is set to $max_g = 50$.

The obtained results are summarized in Table 5, where "Time SW" is the time used by the standalone wrapper method to select the features in [s]. Comparing with the results obtained by exhaustive search (Table 4), it can be highlighted that the selected sensors are the same pair $\{13, 27\}$ and that the average computing time is reduced from 571 to 60 seconds.

Finally, the proposed hybrid method has been tested. The values used for the thresholds has been empirically chosen according to some previous test and with the goal of banning some combinations but still allowing a rich number of tested combinations, those are $\sigma = \alpha/2$ and $\gamma = \alpha/4$. The obtained results are summarized in Table 6. In this table, "Time F" refers to the time of filter computation in [s], "Time W" refers to the time of wrapper computation in [s] and $n_f^{(R)}$ is the number of features that pass the filter. It can be observed that the selected pair of sensors is still the same $\{13, 27\}$. With respect to the computing time, two aspects can be highlighted: First, the computing time of the filter is negligible with respect to the one of the wrapper (in average, 0.032 seconds versus 37 seconds). Second, the filter helps the wrapper to be faster; in particular, a decrease from 60 to 37 seconds in the average computing time is obtained.

Table 5: Results of the Wrapper method in the Hanoi WDN network.

Data set	Sensors	Time W	ATD
1	14, 28	111	1.21
2	11, 29	60	1.26
3	9, 15	49	1.39
4	26, 27	35	1.62
5	10, 27	57	1.16
6	11, 27	66	1.18
7	10, 29	69	1.17
8	**13, 27**	**42**	**1.15**
9	11, 15	55	1.25
10	13, 23	55	1.32
Average	-	60	1.27

Table 6: Results of the proposed hybrid feature selection in the Hanoi WDN network.

Data set	Sensors	Time F	Time W	$n_f^{(R)}$	ATD
1	11, 15	0.032	37	21	1.32
2	11, 29	0.032	30	21	1.27
3	13, 29	0.032	25	20	1.13
4	13, 29	0.032	33	21	1.13
5	13, 29	0.032	33	20	1.14
6	13, 27	0.032	45	21	1.13
7	10, 27	0.032	27	20	1.14
8	10, 15	0.032	34	21	1.29
9	13, 29	0.032	39	21	1.14
10	**13, 27**	**0.032**	**51**	**21**	**1.12**
Average	-	0.032	37	-	1.18

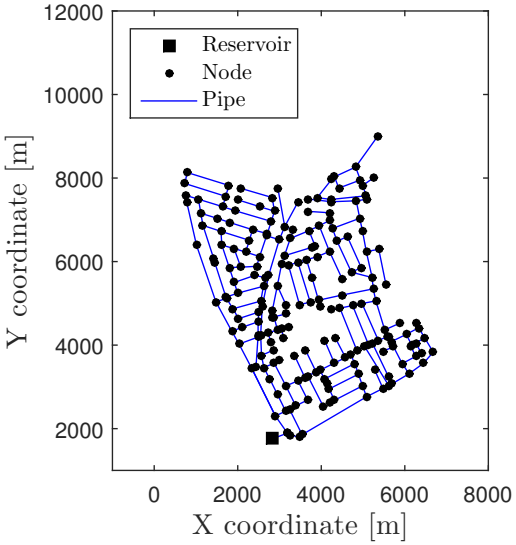

Figure 7: Limassol DMA topological network.

5.2. Limassol DMA case study

The Limassol (Cyprus) DMA network, depicted in Fig. 7, has one reservoir, 197 consumer nodes and 236 pipes. For a network of this size, it is considered realistic to place three pressure sensors (n_s=3). The same uncertainties than the Hanoi WDN case study (except for the leak size, which varies from 4 to 6 [l/s]) are considered to create the data sets, which are again ten complete training+validation data sets and a unique testing data set used to calculate the ATD value showed in Tables 7 and 8. All data sets have the same size as in the Hanoi case study. Finally, the reference input flow pattern shown in Fig. 8 is considered.

For the Limassol DMA network, it is not realistic to apply the Exhaustive Search method. The total time needed to compute and evaluate the performance of all the possible combinations for 197 nodes and select three of them (3,764,670 combinations) can be roughly approximated by assuming that the time of computing each combination, estimated from the average time from 50 combinations, is 240.48 seconds, providing a value of 28.7 years.

As with the Hanoi WDN network, a simplified version of the method that just implements the wrapper has been tested first. The obtained results are summarized in Table 7. Different combinations are selected for the different data sets, but again a further analysis shows that all the combinations are

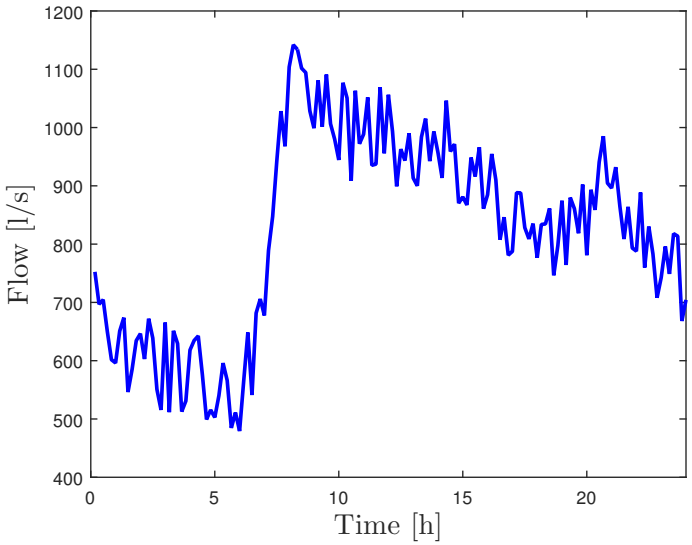

Figure 8: Example of a daily flow consumption in Limassol DMA network.

quite similar since their components are close nodes. The finally selected nodes are the set $\{81, 133, 169\}$. Working with the data set 1 and using the measurements simulated for sensors in these nodes, the obtained ATD is minimal and equal to 3.06. The average value for the ATD for the 10 considered data sets is 3.39. Finally, the value for the average computation time is 18,270 seconds (around five hours).

The results for the hybrid feature selection method are summarized in the Table 8. The more restrictive value $\gamma = \alpha/20$ is used for this example in order to remove a suitable number of features in the filtering stage, while the $\sigma = \alpha/2$ value is maintained. From Table 8, it can be seen that the proposed method performance is better in average than the standalone wrapper method in terms of the objective indicator (ATD) and with a significant reduction in the computation time. The resulting feature selection is depicted in Fig. 9. The selected nodes to place sensors (features) have been the nodes number 40, 152 and 166.

6. Conclusion

In this paper, a sensor placement approach for classifier-based leak localization in water distribution networks using hybrid feature selection is

Table 7: Results of the feature selection using only a wrapper method in Limassol DMA network.

Data set	Sensors	Time W	ATD
1	**81, 133, 169**	**19007**	**3.06**
2	10, 43, 172	24157	3.56
3	28, 110, 170	23773	3.15
4	185, 195, 196	10515	3.64
5	132, 185, 188	13481	3.18
6	87, 189, 190	18200	3.70
7	15, 51, 152	21752	3.51
8	28, 118, 156	16565	3.17
9	87, 189, 190	17526	3.70
10	12, 36, 156	17725	3.31
Average	-	18270	3.39

Table 8: Results of the proposed hybrid feature selection in Limassol DMA network.

Data set	Sensors	Time F	Time W	$n_f^{(R)}$	ATD
1	82, 120, 154	1.64	15828	85	3.09
2	120, 151, 188	1.61	8834	85	3.09
3	**40, 152, 166**	**1.67**	**5429**	**87**	**3.00**
4	40, 194, 197	1.57	9495	87	3.43
5	38, 120, 159	1.59	25178	86	3.02
6	126, 133, 172	1.59	7177	87	3.03
7	38, 103, 172	1.54	11786	87	3.20
8	133, 194, 197	1.90	6174	87	3.50
9	7, 154, 190	1.61	9323	87	3.19
10	40, 154, 190	1.63	12790	87	3.26
Average	-	1.65	11201	-	3.18

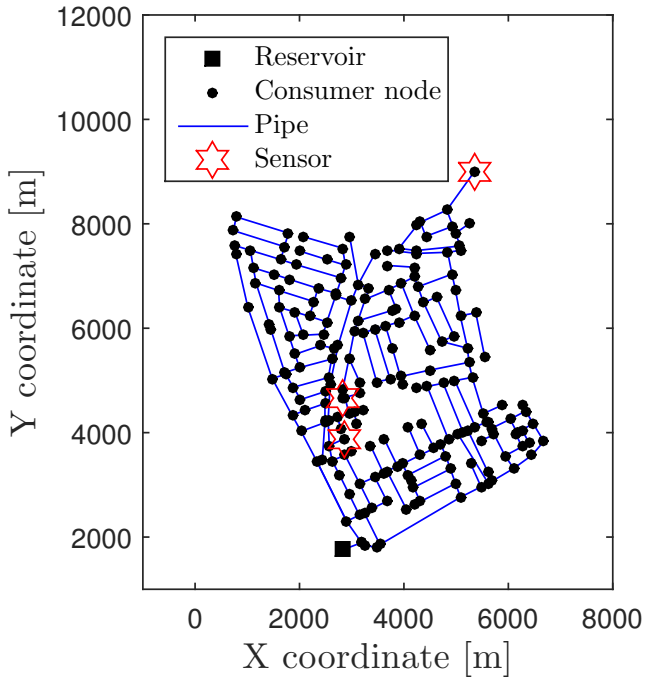

Figure 9: Feature selection as sensor placement in the Limassol DMA network.

presented. The obtained sensor placement based on the feature selection is optimal in the sense that it maximizes the leak isolability when it is used in combination with a classifier-based leak localization method. In order to deal with the combinatory problem in feature selection, the use of a relevance and redundancy filter is proposed in conjunction with the use of genetic algorithms. The performance of the proposed method has been illustrated by means of the application to the Hanoi WDN and a DMA of the Limassol WDN.

Acknowledgments

This work has been funded by the Ministerio de Economía, Industria y Competitividad (MEICOMP) of the Spanish Government through the project DEOCS (ref. DPI2016-76493) by MEICOMP and FEDER, through the grant IJCI-2014-20801 and by the Catalan Agency for Management of University and Research Grants (AGAUR), the European Social Fund (ESF) and the Secretary of University and Research of the Department of Companies and Knowledge of the Government of Catalonia through the grants FI-DGR 2015 (ref. 2015 FI_B 00591).

References

[1] R. Puust, Z. Kapelan, D. A. Savić, T. Koppel, A review of methods for leakage management in pipe networks, Urban Water Journal 7 (1) (2010) 25–45.

[2] M. Mulholland, A. Purdon, M. A. Latifi, C. Brouckaert, C. Buckley, Leak identification in a water distribution network using sparse flow measurements, Computers & Chemical Engineering 66 (2014) 252 – 258.

[3] Y. Kim, S. J. Lee, T. Park, G. Lee, J. C. Suh, J. M. Lee, Robust leak detection and its localization using interval estimation for water distribution network, Computers & Chemical Engineering 92 (2016) 1 – 17.

[4] Z. Y. Wu, P. Sage, Water loss detection via genetic algorithm optimization-based model calibration, in: Systems Analysis Symposium, ASCE, 2006, pp. 1–11.

[5] J. Mashford, D. de Silva, D. Marney, S. Burn, An approach to leak detection in pipe networks using analysis of monitored pressure values by support vector machine, in: Third International Conference on Network and System Security, 2009, pp. 534–539.

[6] A. Soldevila, J. Blesa, S. Tornil-Sin, E. Duviella, R. M. Fernandez-Canti, V. Puig, Leak localization in water distribution networks using a mixed model-based/data-driven approach, Control Engineering Practice 55 (2016) 162 – 173.

[7] A. Soldevila, R. M. Fernandez-Canti, J. Blesa, S. Tornil-Sin, V. Puig, Leak localization in water distribution networks using Bayesian classifiers, Journal of Process Control 55 (2017) 1–9.

[8] D. Wachla, P. Przystalka, W. Moczulski, A method of leakage location in water distribution networks using artificial neuro-fuzzy system, IFAC-PapersOnLine 48 (21) (2015) 1216 – 1223.

[9] V. Rico-Ramirez, S. Frausto-Hernandez, U. M. Diwekar, S. Hernandez-Castro, Water networks security: A two-stage mixed-integer stochastic program for sensor placement under uncertainty, Computers & Chemical Engineering 31 (5âĂŞ6) (2007) 565 – 573.

[10] N.-B. Chang, N. Prapinpongsanone, A. Ernest, Optimal sensor deployment in a large-scale complex drinking water network: Comparisons between a rule-based decision support system and optimization models, Computers & Chemical Engineering 43 (2012) 191 – 199.

[11] R. Mukherjee, U. M. Diwekar, A. Vaseashta, Optimal sensor placement with mitigation strategy for water network systems under uncertainty, Computers & Chemical Engineering 103 (2017) 91 – 102.

[12] R. Sarrate, J. Blesa, F. Nejjari, J. Quevedo, Sensor placement for leak detection and location in water distribution networks, Water Science and Technology: Water Supply 14 (5) (2014) 795–803.

[13] M. V. Casillas, V. Puig, L. E. Garza-Castañón, A. Rosich, Optimal Sensor Placement for Leak Location in Water Distribution Networks Using Genetic Algorithms, Sensors 13 (11) (2013) 14984–15005.

[14] M. À. Cugueró-Escofet, V. Puig, J. Quevedo, Optimal pressure sensor placement and assessment for leak location using a relaxed isolation index: Application to the barcelona water network, Control Engineering Practice 63 (2017) 1 – 12.

[15] J. Blesa, F. Nejjari, R. Sarrate, Robust sensor placement for leak location: analysis and design, Journal of Hydroinformatics 18 (1) (2016) 136–148.

[16] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of machine learning research 3 (Mar) (2003) 1157–1182.

[17] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, bioinformatics 23 (19) (2007) 2507–2517.

[18] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowledge and information systems 34 (3) (2013) 483–519.

[19] J. R. Vergara, P. A. Estévez, A review of feature selection methods based on mutual information, Neural computing and applications 24 (1) (2014) 175–186.

[20] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers and Electrical Engineering 40 (1) (2014) 16–28.

[21] A. Salmerón, A. L. Madsen, F. Jensen, H. Langseth, T. D. Nielsen, D. Ramos-López, A. M. Martínez, A. Masegosa, Parallel filter-based feature selection based on balanced incomplete block designs, in: Proceedings of the European Conference on Artificial Intelligence (ECAI 2016), 2016.

[22] T. Liu, H. Wei, K. Zhang, W. Guo, Mutual information based feature selection for multivariate time series forecasting, in: Control Conference (CCC), 2016 35th Chinese, IEEE, 2016, pp. 7110–7114.

[23] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of machine learning research 5 (Oct) (2004) 1205–1224.

[24] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on pattern analysis and machine intelligence 27 (8) (2005) 1226–1238.

[25] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, Expert systems with applications 41 (4) (2014) 2052–2064.

[26] B. Xue, M. Zhang, W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE transactions on cybernetics 43 (6) (2013) 1656–1671.

[27] R. Díaz-Uriarte, S. A. De Andres, Gene selection and classification of microarray data using random forest, BMC bioinformatics 7 (1) (2006) 3.

[28] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626.

[29] H. H. Inbarani, A. T. Azar, G. Jothi, Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis, Computer methods and programs in biomedicine 113 (1) (2014) 175–185.

[30] Z. Hu, Y. Bao, T. Xiong, R. Chiong, Hybrid filter-wrapper feature selection for short-term load forecasting, Engineering Applications of Artificial Intelligence 40 (2015) 17 – 27.

[31] J. Apolloni, G. Leguizamón, E. Alba, Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments, Applied Soft Computing 38 (2016) 922–932.

[32] L. Ferrandez-Gamot, P. Busson, J. Blesa, S. Tornil-Sin, V. Puig, E. Duviella, A. Soldevila, Leak localization in water distribution networks using pressure residuals and classifiers, IFAC-PapersOnLine 48 (21) (2015) 220 – 225.

[33] R. Sarrate, J. Blesa, F. Nejjari, Clustering techniques applied to sensor placement for leak detection and location in water distribution networks,

in: Control and Automation (MED), 2014 22nd Mediterranean Conference of, 2014, pp. 109–114.

[34] K. Deep, K. P. Singh, M. L. Kansal, C. Mohan, A real coded genetic algorithm for solving integer and mixed integer optimization problems, Applied Mathematics and Computation 212 (2) (2009) 505–518.

[35] I. S. Oh, J. S. Lee, B. R. Moon, Hybrid genetic algorithms for feature selection., IEEE transactions on pattern analysis and machine intelligence 26 (11) (2004) 1424–1437.

[36] P. Cugueró-Escofet, J. Blesa, R. Pérez, M. À. Cugueró-Escofet, G. Sanz, Assessment of a leak localization algorithm in water networks under demand uncertainty, IFAC-PapersOnLine 48 (21) (2015) 226 – 231.

[37] A. Soldevila, R. M. Fernandez-Canti, J. Blesa, S. Tornil-Sin, V. Puig, Leak localization in water distribution networks using Bayesian classifiers, in: 2016 European Control Conference, ECC 2016, 2016, pp. 1758–1763.

[38] A. Soldevila, S. Tornil-Sin, R. M. Fernandez-Canti, J. Blesa, V. Puig, Optimal sensor placement for classifier-based leak localization in drinking water networks, in: Proceedings of the 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol), 2016, pp. 319–324.